

Remarks

Claims 1-8, 10-28 and 30-42 are pending in the application. Claims 1, 21, 41 and 42 are independent.

Claims 1-8, 10-28 and 30-42 have been rejected.

Claim 42 is amended to more precisely be directed to a computer program product comprising a computer readable memory, in rectification of the 35 USC 101 rejection. Support for this amendment may be found at least at page 9 of the specification as filed.

Claims 1, 21, 41 and 42 are amended to reflect more clearly that the provisioning instructions are embedded in the content of the provisioning application. Support for this amendment may be found at pages 15 and 16 of the specification as filed, under sections entitled "Application 107 Intelligence" and "Provisioning Configuration Data" respectively therein.

No new matter is added by way of the claim amendments.

Double Patenting

Claims 1, 21, 41 and 42 were rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-3, 8, 11, 15 and 29 of U.S. Patent No. 7,509,658.

Attached herewith, Applicant submits a terminal disclaimer in compliance with 37 CFR 1.32c, in view of the fact that the conflicting patent is commonly owned with this application. Applicant hereby considers the nonstatutory obviousness-type double patenting as overcome.

Claim Rejections 35 USC 103

Applicant thanks the Examiner for laying out the detailed explanations in support of the Examiner's claim rejections based on 35 USC 103 in the Office Action dated 06/03/2010.

The Examiner rejected claims 1-8, 10-28 and 30-42 under 35 U.S.C. 103(a) as being anticipated by Jensen (US 2004/0261086) in view of Kjellberg and Krantz. The terminology of the Office Action with regard to the cited references is adopted herein. Applicant respectfully traverses the rejections in view of the arguments herein.

Exemplary claim 1 as amended herein recites: *A method for providing customized provisioning of an application on a runtime environment of a terminal, the application including content having at least one content type, the method comprising the steps of:*

obtaining the content by the runtime environment;

for each content type, obtaining by the runtime environment a set of provisioning instructions related to the content type, the provisioning instructions being customized by distributed provisioning control through the provisioning instructions for different versions of the application, the provisioning instructions embedded within content of the application for specifying a provisioning application program interface (API) set for provisioning the content on the terminal; and

executing by the runtime environment the provisioning instructions for employing the API set, by a script interpreter, to provision the application according to the specified content type.

A. None of the cited references, including Jensen, disclose the claimed feature of “the provisioning instructions embedded within content of the application for specifying a provisioning application program interface (API) set for provisioning the content on the terminal”

The Examiner asserted at page 8 of the Office Action that Jensen disclosed this feature (prior to the amendment herein), citing Fig. 3 and paragraphs [0028]-[0031], [0036]-[0039] and [0041]-[0043] of Jensen.

Based on careful review of the cited paragraphs of Jensen, Applicant respectfully traverses the Examiner's characterization of Jensen as disclosing the claimed feature (as amended herein). Figure 3 of Jensen, for example, discloses Provisioning API set 222 as including discovery transactions 300, subscription transactions 302 and delivery transactions 304, thus defining the 3 roles a provisioning adapter 206 can play with respect to the provisioning application 208 of Jensen.

However, there is no teaching of “the provisioning instructions embedded within content of the application for specifying a provisioning application program interface (API) set for provisioning the content on the terminal”.

In contrast, Applicant's specification as filed, at page 16 describes this feature:

Provisioning Configuration Data

Provisioning Data consists of an arbitrary set of properties, which can be referred to as an embedded or otherwise coupled version of the provisioning instructions 124, that may be specified in the content of the application 107. This approach to application 107 intelligence relies on the fact that the Framework 206 recognizes the set of possible properties and modifies the provisioning process to suit customized settings. It is recognised that the provisioning data could be designed as external to the content of the application 107.

Having the provisioning instructions coupled to the application, as claimed in claim 1, allows the same application to be provisioned in a manner commensurate with the different hardware capabilities of different client devices, doing so to take fully exploit device-specific capabilities such as screen display resolution and processor speed, among others, even if the different devices employ the same operating system.

Thus Jensen, nor any of the other cited references, disclose the claimed feature of “*the provisioning instructions embedded within content of the application for specifying a provisioning application program interface (API) set for provisioning the content on the terminal*”.

B. None of the cited references, including Kjellberg, discloses “the provisioning instructions being customized for different subsets of versions of the application”.

The Examiner, at page 9 of the Office Action of 06/03/2010, acknowledges that Jensen does not disclose the former-claim 1 element of “*the provisioning instructions being customized for different subsets of versions of the application*”, and maintains that Kjellberg at paragraphs [0024] to [0026] discloses that claim element.

The Examiner cited Kjellberg at paragraphs [0024] to [0026] as disclosing “*the provisioning instructions being customized for different subsets of versions of the application*”. Applicant respectfully disagrees with such characterization.

Kjellberg at paragraph [0025] provides:

[0025] With reference now to FIG. 1 of the drawings, there is illustrated a provisioning server 200 capable of provisioning objects and applications to client devices 100 in real-time. As noted above, provisioning is the capability to receive a request for an application or object, find a suitable version of the requested application or object and provide the application or object to the requester. The ability to find a suitable version of the requested application or object accounts for the different formats utilized by the many different types of client devices 100, each with its own characteristics, limitations and configuration. For example, the client devices 100 may include PDAs 100a, workstations and desktop computers 100b, mobile phones 100c and laptops 100d. The characteristics and configurations of each

Nowhere in Kjellberg is there disclosed the feature and structure for customized provisioning as disclosed in the subject matter of claim 1. Rather, Kjellberg discloses a static provisioning procedure that is implemented by the client devices 100, and which is bound by the different types of client devices 100 .
"The ability to find a suitable version of the requested application or object accounts for the different formats utilized by the many different types of client devices 100" from Kjellberg paragraph [0025].

In contrast, claim 1 provides the feature: *"for each content type, obtaining by the runtime environment a set of provisioning instructions related to the content type, the provisioning instructions being customized by distributed provisioning control through the provisioning instructions for different versions of the application"*.

As disclosed in the specification of the subject application as filed, at page 10-11:

The processing framework 206 manages the application 107 provisioning, retrieving and removing from the runtime environment of the terminal 100. The processing framework 206 provides ability to dynamically manage the provisioning API 122. It is recognized that the control of provisioning the content of the application 107 is distributed between the application 107 through the provisioning instructions 124 and the provisioning service 308 of the Framework 206, which represents the separation as Application 107 Intelligence and Framework 206 Intelligence. Having made this separation, the application 107

may no longer be bound by static provisioning procedures that would normally be imprinted on terminal 100 prior provisioning procedures.

Thus the feature of customized provisioning is described in the subject application as one where provisioning is no longer bound by static provisioning procedures that would normally be bound by the terminal provisioning procedures.

Furthermore, that customized provisioning is achieved by distributing the provisioning control between the application 107 (through the provisioning instructions) and the framework 206. From page 14 of the specification as filed:

Customized Provisioning

To provide customized provisioning capability, the provisioning control of the application 107 on the terminal 100 is distributed between the application 107 (through the provisioning instructions 124) and the framework 206. This separation of control as Application 107 provisioning control (Intelligence) and Framework 206 provisioning control (Intelligence) helps the application 107 to be no longer bound by a static provisioning procedure that would normally be implemented by current terminals 100.

Applicant therefore respectfully maintains that Kjellberg does not disclose the claim 1 limitation of "*the provisioning instructions being customized for different subsets of versions of the application*", and further submits that any reasonable interpretation of the cited passages of Kjellberg would be consistent such conclusion.

Independent claims 21, 41 and 42 are similar in scope to claim 1, and therefore a similar argument applies. Accordingly, we submit that the rejection to these claims be withdrawn for at least the same reasons.

Since the remaining dependent claims depend from one of the above noted independent claims, Applicant submits that the rejection of these claims be withdrawn for at least the same reasons.

For the foregoing reasons, the Applicant respectfully submits that the claimed invention is patentable over the prior art. Reconsideration and allowance of the claims is respectfully requested.

Respectfully submitted,

/HenryOhab/

Henry Ohab
Registration No. 45,854
ph: (416) 862-3593

Gowling Lafleur Henderson LLP
1 First Canadian Place, Suite 1600
Toronto, Ontario, M5X 1G5

TOR_LAW\17458815\1